

Name: Satyam Rai

UID: 2019130051

Branch: TE COMPS – C

Subject: AI/ML

Aim: To implement the Tic-Tac-Toe game using the A* search strategy.

Code:

```
winCondition = [
    [(0, 0), (0, 1), (0, 2)],
    [(1, 0), (1, 1), (1, 2)],
    [(2, 0), (2, 1), (2, 2)],
    [(0, 0), (1, 0), (2, 0)],
    [(0, 1), (1, 1), (2, 1)],
    [(0, 2), (1, 2), (2, 2)],
    [(0, 0), (1, 1), (2, 2)],
    [(0, 2), (1, 1), (2, 0)],
]

#Board Environment
env = [
    ['_', '_', '_'],
    ['_', '_', '_'],
    ['_', '_', '_'],
]

finalCount = 5

def calc_F_Value(i, j):
    maxElement = [-1, -1, -1]
    for _ in winCondition:
        empty = 0
        dot = 0
        cross = 0
        if (i, j) in _:
            for k in _:
                if env[k[0]][k[1]] == '_':
```

```

        empty += 1
        if env[k[0]][k[1]] == 'o':
            dot += 1
        if env[k[0]][k[1]] == 'x':
            cross += 1
    if maxElement[2] < cross:
        maxElement = [i, j, cross]

    return maxElement

def playAI():
    fvalues = []
    for i in range(3):
        for j in range(3):
            if env[i][j] == '_':
                env[i][j] = 'o'
                fvalues.append(calc_F_Value(i, j))
                env[i][j] = '_'
    position = max(fvalues, key=lambda x: x[2])
    env[position[0]][position[1]] = 'o'

def currStatus():
    flagH = None
    counter = 0
    for i in range(3):
        for j in range(3):
            if env[i][j] != '_':
                counter += 1

    if counter == 9:
        flagH = "Draw"

    for location in winCondition:
        if env[location[0][0]][location[0][1]] == 'x' and
env[location[1][0]][location[1][1]] == 'x' and
env[location[2][0]][location[2][1]] == 'x':
            flagH = True
            break
        elif env[location[0][0]][location[0][1]] == 'o' and
env[location[1][0]][location[1][1]] == 'o' and
env[location[2][0]][location[2][1]] == 'o':
            flagH = False
            break
    return flagH

```

```

def print_env():
    print(env[0][0], "|", env[0][1], "|", env[0][2])
    print(env[1][0], "|", env[1][1], "|", env[1][2])
    print(env[2][0], "|", env[2][1], "|", env[2][2])
    print("\n\n")

endFlag = False

print_env()

while True:
    #take the current location input
    playerLoc = list(map(int, input("Enter your next move location:
").strip().split()))
    playerLoc = [playerLoc[0] , playerLoc[1] ]

    if env[playerLoc[0]][playerLoc[1]] != '_':
        print("!!It's not an empty cell!!")
        continue

    env[playerLoc[0]][playerLoc[1]] = 'x'
    print_env()

    gameStatus = currStatus()
    if gameStatus == True:
        print("You won!!")
        endFlag = True
        break
    elif gameStatus == False:
        print("You lost!!")
        endFlag = True
        break
    elif gameStatus == "Draw":
        print("Match Draw!!")
        endFlag = True
        break

    if not endFlag: playAI()

    print_env()

    gameStatus = currStatus()
    if gameStatus == True:

```

```
        print("You won!!")
        break
    elif gameStatus == False:
        print("You lost!!")
        break
    elif gameStatus == "Draw":
        print("Match Draw!!")
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\vinay\Desktop\AIML\Exp 3> python -u "c:\Users\vinay\Desktop\AIML\Exp 3\tictactoe.py"

_ | _ | _
_ | _ | _
_ | _ | _

Enter your next move location: 0 0
x | _ | _
_ | _ | _
_ | _ | _

x | o | _
_ | _ | _
_ | _ | _

Enter your next move location: 2 0
x | o | _
_ | _ | _
x | _ | _

x | o | _
o | _ | _
x | _ | _
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Enter your next move location: 2 2
x | o | _
o | _ | _
x | _ | x

x | o | _
o | o | _
x | _ | x

Enter your next move location: 2 1
x | o | _
o | o | _
x | x | x

You won!!
```

Conclusion:

In this experiment, I have implemented the Tic-Tac-Toe game in python using A* search strategy. Here the player uses 'X' while the agent (AI) uses 'O' to play the game. The position of a player's move is used to calculate the heuristic value. If all the possible cells of the board are filled without a valid combination of symbols, there is a "Draw". Else, either one of the player or the AI wins the game depending on a valid combination of 'X' or 'O'.